

数据结构实验：查找和排序

原创

[basement_boy](#) 于 2014-06-03 22:12:36 发布 3661 收藏 22

分类专栏: [数据结构](#) 文章标签: [数据结构](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/u012278856/article/details/28303059>

版权



[数据结构](#) 专栏收录该内容

13 篇文章 0 订阅

订阅专栏

数据结构课的实验, 小组中给我分了这个。。。

大一在石河子大学念的, 那时一到写代码或是做课程设计时大多同学都是直接从网上找, 然后就让我和几个兄弟

如果这篇东西荣幸的被你看到, 而你也恰是华东理工大学计算机专业的, 打算应付这个实验的话, 建议复制走后

查找和排序

一、实验目的

1. 掌握顺序查找，二分法查找，分块查找的算法。
2. 掌握各种排序算法及其性能的比较

二、实验内容

1. 编写一个程序输出在顺序表{13, 22, 35, 43, 54, 68, 71, 82, 98, 1005}
<http://blog.csdn.net/u012278856>
中采用顺序方法和折半方法查找某个关键字的过程。
2. 编写一个程序实现直接插入排序过程，并输出{94, 28, 57, 66, 35, 84, 63, 42, 71, 10}的排序过程
3. 编写一个程序实现快速排序算法，并输出{94, 28, 57, 66, 35, 84, 63, 42, 71, 10}的排序过程。

直接贴代码：

第一个：

```
///2014.6.3

/*编写一个程序输出在顺序表
{13, 22, 35, 43, 54, 68, 71, 82, 98, 1005}
中采用顺序方法和折半方法查找某个关键字的过程。*/

#include <iostream>
#include <cstdio>
using namespace std;

//顺序查找函数
//date是要查找的数组，x是要在此数组中查找的数，n是数组大小。
bool ordersearch(int date[],int x,int n){
    for(int i=0 ; i<n ; i++){
        if( date[i] != x )
            continue;
        if( date[i] == x )
            return true;
    }
    return false;
}
```

```

    cout<<"第 "<<l+1<<" 个数不是要查找的数"<<endl;
else{
    cout<<"第 "<<i+1<<" 个数是要查找的数,下标是"<<i<<endl;
    return true;
}
}
cout<<"查找失败, 该数不在数组中"<<endl;
return false;
}

//折半查找函数
//date是要查找的数组, x是要在此数组中查找的数, n是数组大小。
bool binsearch(int date[],int x,int n){
    int l,r;
    l = 0;
    r = n-1;
    int mid = (l+r)/2;
    while( l != r ){
        if( date[l] == x ){
            cout<<"第 "<<l+1<<" 个数是要查找的数,下标是"<<l<<endl;
            return true;
        }
        if( date[r] == x ){
            cout<<"第 "<<r+1<<" 个数是要查找的数,下标是"<<r<<endl;
            return true;
        }
        mid = (l+r)/2;
        if( date[mid] == x ){
            cout<<"第 "<<mid+1<<" 个数是要查找的数,下标是"<<mid<<endl;
            return true;
        }
        else if( date[mid] < x ){
            cout<<"要查找的数不在"<<l<<","<<mid<<">中"<<endl;
            l = mid;
        }
        else{
            cout<<"要查找的数不在"<<mid<<","<<r<<">中"<<endl;
            r = mid;
        }
    }
    cout<<"查找失败, 该数不在数组中"<<endl;
    return false;
}

int main(){
    int date[] = {13,22,35,43,54,68,71,82,98,1005};

    cout<<"顺序查找:"<<endl<<endl;
    ordersearch(date,98,10);

    cout<<endl<<"折半查找:"<<endl<<endl;
    binsearch(date,98,10);

    return 0;
}

```

第二个:

```

///2014.6.3

/*编写一个程序实现直接插入排序过程,
并输出{94,28,57,66,35,84,63,42,71,10}的排序过程*/

#include <iostream>
#include <cstdio>
#include <string>
using namespace std;

//该函数用于显示排序过程中数组中的数据变化
void showdetail(int date[],int n,int i){
    cout<<"第 "<<i<<" 轮:"<<endl;
    cout<<"有序序列:";
    for(int j=0 ; j<i ; j++)
        cout<<" "<<date[j];
    cout<<endl;
    cout<<"无序序列:";
    for(int j=i ; j<n ; j++)
        cout<<" "<<date[j];
    cout<<endl<<endl;
}
//该函数用于显示最终排序结果
void showresult(int date[],int n){
    cout<<"最终结果是: "<<endl;
    for(int i=0 ; i<n ; i++){
        cout<<date[i]<<" ";
    }
    cout<<endl;
}

//直接插入排序函数
//date数组是要排序的数组, n是数组的元素个数
void insertsort(int date[],int n){
    for(int i=1 ; i<n ; i++){
        showdetail(date,n,i);
        int temp = date[i];
        for(int j=0 ; j<i ; j++){
            if( date[j]>date[i] ){
                for(int p=i ; p>j ; p--){
                    date[p] = date[p-1];
                }
                date[j] = temp;
            }
        }
    }
    showresult(date,n);
}

int main(){
    int date[] = {94,28,57,66,35,84,63,42,71,10};

    cout<<"直接插入排序:"<<endl<<endl;
    insertsort(date,10);

    return 0;
}

```

第三个:

```
///2014.6.3

/*编写一个程序实现快速排序算法,
  并输出{94, 28, 57, 66, 35, 84, 63, 42, 71, 10}的排序过程*/

#include <iostream>
using namespace std;

//该函数用于显示每次调用Qsort()后的排序结果
//其中用()括起来的该次排序的枢纽元素
void show(int date[],int n,int p){
    static int num = 0;
    num++;
    cout<<"下面是第"<<num<<"次调用Qsort()处理的情况:"<<endl;
    for(int i=0 ; i<p ; i++)
        cout<<date[i]<<" ";
    cout<<"("<<date[p]<<") ";
    for(int i=p+1 ; i<n ; i++)
        cout<<date[i]<<" ";
    cout<<endl;
}
//快速排序函数,date是要排序数组, n是数组元素个数
void Qsort(int date[],int n){
    if( n<=1 )
        return;
    int p = 0,q = n-1;
    int k = date[0];
    while( p<q ){
        while( p<q && date[q]>=k )
            q--;
        date[p] = date[q];
        while( p<q && date[p]<=k )
            p++;
        date[q] = date[p];
    }
    date[p] = k;
    show(date,n,p);
    Qsort(date,p);
    Qsort(date+p+1,n-1-p);
}

int main(){
    int date[] = {94,28,57,66,35,84,63,42,71,10};

    cout<<"快速排序:"<<endl;
    Qsort(date,10);

    cout<<"最终结果:"<<endl;
    for(int i=0 ; i<10 ; i++){
        cout<<date[i]<<" ";
    }
    cout<<endl;
    return 0;
}
```